Département de Physique

Département de Physique



UNIVERSITÉ IBN TOFAIL
FACULTÉ DES SCIENCES
DÉPARTEMENT DE PHYSIQUE

Mémoire de Projet de Fin d'Etudes

Filière : SMP

Programation

exosup.com

Élaboré par:

Bilal BELMOKHTAR
Chaimae EL-LICHA
Mohammed NEGGOUR

Encadré par:Pr. Ibrahimi

Pr. Youssef El merabet

Soutenu le 20 Mai 2015 devant le Jury :

yellownumer Prenom et Nom	*Président*
Prenom et Nom	*Encadrant*
Prenom et Nom	*Examinateur*

Année Universitaire : 2014 2015 –

P
F
E

exosup.com
page facebook

Remerciements

Nous tenons à remercier M. Youssef El merabet, pour avoir dirigé et guidé notre PFE et pour nous avoir proposé ce sujet de recherche…

Nous remercions tout d'abord M. Nom et Prenom et M. Nom et Prenom, Professeurs à la Faculté des Sciences de Kenitra pour avoir gentiment accepté de juger ce travail et participé à notre jury.…

…

UNIVERSITÉ IBN TOFAIL
Département de Physique

Programation

Résumé :

D'un point de vue général, les travaux de recherche de ce projet de fin d'études...

Mots-clés : Mot1, Mot2, ...

Abstract :

The work presented in this "final year study project" is developed in a global approach that consists in ...

Keywords : Mot1, Mot2, ...

exosup.com page facebook

SOMMAIRE

Remerciements **1**

Abstract **2**

Liste des Figures **5**

Liste des Tables **6**

1 Analyse numérique **1**
- 1.1 Introduction 1
- 1.2 Généralités 1
- 1.3 Définition 1
  - 1.3.1 Les objectifs d'Analyse Numérique 2
  - 1.3.2 Relation entre l'analyse numérique et les méthodes numériques . . 2
- 1.4 Exemple d'analyse numérique : Réesolution d'un système d'équation . . 2
  - 1.4.1 Résolution d'un système d'équation non linaire 2
  - 1.4.2 Résolution d'un système d'équation linaire 3

2 Énoncées des Programmations Numériques **5**
- 2.1 Représentation de programme (algorithme) 5
- 2.2 Organigramme 8
  - 2.2.1 Constantes et Variables 8
  - 2.2.2 Expression 9
  - 2.2.3 Instructions 11
    - 2.2.3.1 Traitements conditionnels 12
    - 2.2.3.2 Boucles 13

3 Applications aux instruments d'optiques **25**
- 3.1 Structure d'un oeil normal 25
- 3.2 Les défauts du système optique 27
  - 3.2.1 A- L'oeil myope et l'oeil hypermétrope 27
    - 3.2.1.1 B- L'oeil astigmate 29
    - 3.2.1.2 C- Problèmes de transparence 31
- 3.3 Les défauts d'adaptation 31
  - 3.3.1 A- L'oeil presbyte 31
  - 3.3.2 B- Lorsque vos "lunettes de Soleil" n'en sont pas 32
- 3.4 Les problèmes de la rétine 32

3.4.1 A- Le daltonisme 32
3.4.2 C- Les limites inhérentes à la nature de la rétine 33

A Manuel de Latex **34**
A.1 Introduction 34
A.2 Insértion des caractères spéciaux 34
A.3 Les Listes 35
A.3.1 Listes simples 35
A.3.2 Listes numérotées 35
A.4 Les Tableaux 35
A.4.1 Exemple simple d'un tableau avec colorisation des cellules: 36
A.4.2 Exemple d'un tableau avec fusion de cellules 36
A.4.3 Exemple d'un tableau avec changement de la taille des cellules . . 37
A.5 Les Figures 37
A.5.1 Insértion d'une simple figure 38
A.5.2 Insértion des figures dans un tableau 38
A.5.3 Placement des figures côte à côte 38
A.6 Les Equations 38
A.6.1 fonction 41


exosup.com

exosup.com page facebook

Liste des Figures

2.2 6
2.3 7
2.4 7
2.5 8
2.7 10
2.8 11
2.9 12
2.10 13
2.12 15
2.13 17
2.14 18
2.15 19
2.16 20
2.17 21
2.18 23

3.1 26
3.2 Un oeil myope forme l'image d'un objet très éloigné avant la rétine . . . 27
3.3 Un oeil hypermétrope, s'il n'accommode pas, forme l'image d'un objet très éloigné après la rétine 28
3.4 29

A.1 Logo de Ibn Tofail 38
A.2 Insertion des figures dans un tableau 38
A.3 Exemple 2 39
A.4 Placement des figures côte à côte avec l'instruction minipage 39

Liste des Tables

A.1 Ecrire Votre légende ici. 36
A.2 Ecrire Votre légende ici. 36
A.3 Mean value of VINET criterion. 37
A.4 Votre légende ici. 37
A.5 Votre légende ici. 37

Chapitre 1

Analyse numérique

1.1 Introduction

Votre Texte Ici.

1.2 Généralités

L'analyse numérique traite de nombreux problèmes de simulations ou d'expérimentations mathématiques, sciences physiques, biologiques, technologiques ou des problèmes issus de modèles économiques et sociaux. Elle intervient dans le développement de codes de calcul. Elle entretient des liens étroits avec l'informatique. Si sa partie théorique relève plus des mathématiques, sa mise en pratique aboutit généralement à l'implémentation d'algorithmes sur ordinateur. Ses méthodes se fondent à la fois sur la recherche de solutions exactes comme dans le cas de l'analyse matricielle ou du calcul symbolique, sur des solutions approchées qui résultent le plus souvent de processus de discrétisation comme dans le traitement des équations différentielles.

1.3 Définition

L'analyse numérique

L'analyse numérique est l'étude des algorithmes permettant de résoudre les problèmes de mathématiques. Elle s'intéresse aux fondements théoriques et à la mise en pratique des méthodes permettant de résoudre, par des calculs purement numériques, des problèmes d'analyse mathématique.

méthode numérique

Méthode numérique est un développement d'un algorithme Un algorithme numérique décompose un problème en plusieurs étapes où chacune d'elle peut facilement être interprété par l'ordinateur

1.3.1 Les objectifs d'Analyse Numérique

Les différents objectifs du traitement numérique sont:
• résoudre numériquement des problèmes complexes qui ne peuvent pas être traités de façon analytique.
• limiter au maximum le temps de calcul.
• réduire au maximum les erreurs.

1.3.2 Relation entre l'analyse numérique et les méthodes numériques

La méthode numérique est un calcul permettant de transmettre et donner une instruction qui servir et à réduire les étapes de calcule mathématique sur un programme quelconque, et l'analyse numérique simplifier le calcul mathématique sur l'algorithme.

1.4 Exemple d'analyse numérique : Réesolution d'un système d'équation

On utilise l'analyse numérique plus souvent pour résoudre des problèmes mathématiques, ces problèmes peuvent être résolus numériquement de façon exacte par un algorithme en un nombre fini d'opérations.

1.4.1 Résolution d'un système d'équation non linaire

Méthode de Newton
Elle est la plus utilisée pour trouver la racine d'une fonction non linéaire. Elle requiert cependant l'évaluation de f(x) et de f'(x). Si x0 est la valeur approchée de la solution f(x)=0 dans l'intervalle [a,b], on corrige cette solution par x1= x0 + h où h est une petite valeur.
Si on utilise le développement limité de Taylor au 1er ordre, on a :

f(x_0) $= f(x_0 + h) \approxeq x_0 + h.f'(x_0)$

Si

$f(x_1) = 0$

alors

$h = -\dfrac{f(x_0)}{f'(x_0)}$

Donc

$x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$

LeprincipedelaméthodedeNewtonconsisteàchoisirununevaleurarbitrairex0, calculerx1puisx2àpartir

1.4.2 Résolution d'un système d'équation linaire

Méthode du pivot de Gauss

On utilise la méthode du pivot de Gauss pour résoudre un système linéaire de N équations à N inconnues ce qui est parfaitement rigoureuse du point de vue mathématique. Cette méthode est la plus simple et la plus classique, elle correspond exactement à ce que l'on fait :

- De la première équation, on exprime la première inconnue x1 en fonction des autres inconnues, puis on substitue cette expression dans les autres équations du systèmes, faisant apparaître ainsi un sous-système dans lequel x1 n'intervient pas.
- On refait la même chose jusqu'à ce que la matrice du système devienne triangulaire supérieure (toute la partie inférieure de la matrice - sous la diagonale - est remplie de zéros).
- Une fois la triangularisation terminée, on calcule xn, puis xn1, et ainsi de suite jusqu'à x1 ; les différentes solutions sont ainsi calculées de proche en proche.

exemple: Si on veut résoudre un système de 3 équations à 3 inconnus $\{2x + 2y + 4y = 1$

$2x + 10y + 6z = 1$

$x + 8y + 2z = 1$

Considérons le système linéaire

$$\begin{pmatrix} 2 & 2 & 4 \\ 2 & 10 & 6 \\ 1 & 8 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

exosup.com page facebook

$$\begin{pmatrix} 2 & 2 & 4 \\ 0 & 2 & 2 \\ 0 & 4 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1/2 \\ 0 \\ 1/2 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 2 & 4 \\ 0 & 1 & 1 \\ 0 & 0 & -4 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1/2 \\ 0 \\ 1/2 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 2 & 4 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1/2 \\ 0 \\ -1/8 \end{pmatrix}$$ donc:

z= -1/8, donc y = 1/8 et x= 1/4

Chapitre 2

Énoncées des Programmations Numériques

Énoncées des Programmations Numériques

2.1 Représentation de programme (algorithme)

Un programme est un texte constitué d'un ensemble des directives appelées instructions, qui représentent une suite d'actions élémentaires que l'ordinateur doit accomplir sur des données en entrée, afin d'atteindre le résultat recherché. Évolution des langages de programmation:
L'ordinateur ne sait exécuter que qu'un certain nombre limité d'opérations codées en binaire (langage machine).
Les premiers programmes étaient écrits en binaire; C'était une tache très difficile et exposés aux erreurs.
Par la suite, pour faciliter le travail, on a utilisé des codes mnémoniques pour définir les opérations (ADD, DIV, SUB, MOVE . . .) ses symboles sont traduit en langage machine par un programme appelé assembleur.

Les programmes écrits en langages évolués doivent également être traduits en langage machine pour être exécutés. La conversion se fait de deux manières: la compilation et l'interprétation.
La compilation consiste à traduire dans un premier temps l'ensemble du programme en langage machine. Dans un deuxième temps, le programme est exécuté.

L'interprétation consiste à traduire chaque instruction du programme en langage machine avant l'exécuter .

Démarche de programmation:

La façon d'écrire un programme est intimement liée au langage de programmation que l'on a choisi, mais la démarche programmation se déroule généralement en deux étapes:

D'abord on identifie les données du problème, les résultats recherchés et par quel moyen on peut obtenir ces dernier à partir des données. C'est l'étape d'analyse qui aboutit à un procédé de résolution appelé algorithme .

Un algorithme, c'est une suite d'instructions, qui une fois exécutée correctement, conduit au résultat recherché.

Un algorithme doit donc contenir uniquement des instructions compréhensibles par celui qui devra l'exécuter.

Dans un deuxième temps, on traduit dans le langage de programmation choisi l'algorithme établi dans la phase précédente. Si l'algorithme est bien écrit, sans faute logique, l'étape suivante ne doit normalement poser aucun problème conceptuel. C'est une simple traduction que vous devez effectuer.

Structure d'un programme:

Tout programme est toujours constitué de trois phases:

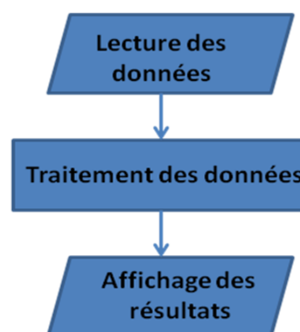


Figure 2.2

Exemple d'une écriture d'un programme: Exemple: un algorithme qui fait la somme de

`#include <STDIO.H>`	←Inclusion des fichiers de bibliothèque.
`#define PI 3.14`	←Déclaration des constantes.
`float r,p;`	←Déclaration des variables globales
`void perimetre(float rayon,float *peri);`	←Déclaration des Prototypes. (Une déclaration de prototype se termine toujours par un point virgule ;).
`void perimetre(float rayon,float *peri)`	←Déclaration des procédures, fonctions ou de Sous Programmes.
`{`	←Début de la procédure
`*peri=2*PI*rayon;`	
`}`	←Fin de la Procédure
`void main()`	←Programme Principal ou ordonnancement.
`{`	←Début du programme Principal
`r=3;`	
`perimetre(r,&p);`	←Instructions
`printf("Le perimetre du cercle de rayon %f est égal à %f",r,p);`	
`}`	←Fin du Programme Principal

FIGURE 2.3

deux entiers.

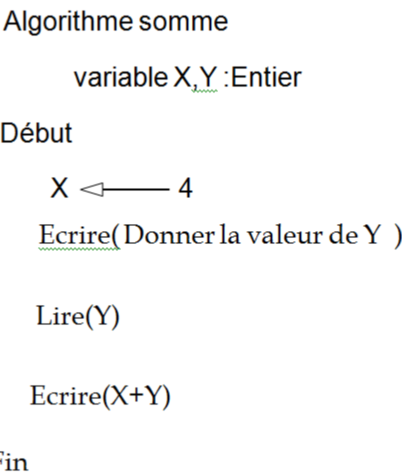

Algorithme somme
variable X,Y :Entier
Début
X ←—— 4
Ecrire(Donner la valeur de Y)
Lire(Y)
Ecrire(X+Y)
Fin

FIGURE 2.4

2.2 Organigramme

2.2.1 Constantes et Variables

Définition d'une constante:
Elle ne change jamais de valeur pendant l'exécution d'un programme. Elle est généralement stockée dans la mémoire morte d'une machine.

Il n'y a pas d'allocation mémoire, mais on peut affecter à un identificateur (Nom) une valeur constante par l'instruction define.
Syntaxe:

define $\langle identificateur \rangle < valeur >$

Exemple: **Définition:** Les variables contiennent les valeurs utilisées pendant l'exécution

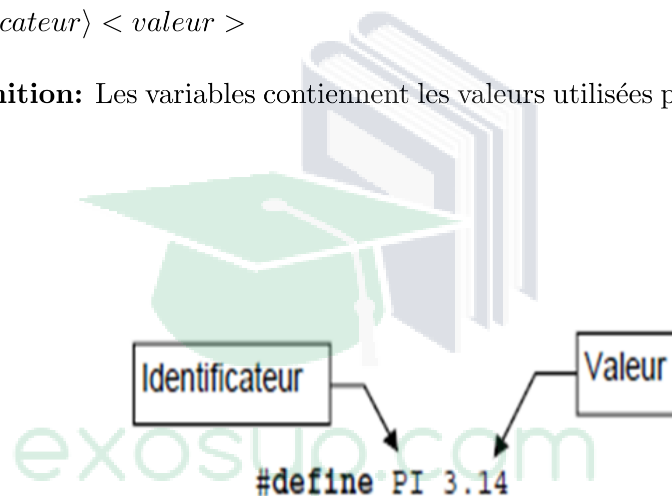


Figure 2.5

du programme.

exosup.com page facebook

Déclarer une variable, c'est prévenir le compilateur qu'un nom va être utilisé pour designer un emplacement de la mémoire.
Les noms des variables sont des identificateurs quelconques.

Toute variable doit être déclarée avant les instructions et son type spécifié dès la déclaration.
Syntaxe:
$\langle type\rangle\langle identificateur1\rangle, .., \langle identificateurn\rangle; Exemple$:

Une variable ou une constante est définie par cinq éléments:
L'identificateur : c'est son nom.
Le type : entier, caractère, chaine de caractère, réel.
La taille : nombre d'octets occupés en mémoire.
La valeur : c'est la valeur que l'on attribue à la variable ou la constante.
L'adresse : c'est l'emplacement mémoire où est stocké la valeur de la variable.
Entier : int I; short S; long L; C'est un nombre positif ou négatif : ex 1584 ou -458.
Caractère : char C; C'est un nombre entier positif compris entre 0 et 255.
Chaine de caractère: char Chaine[10];C'est un ensemble de caractères.
Réel : float F; double DF; C'est un nombre à virgule positif ou négatif avec un exposant ex: 12,45 104.
Boolean : pas de déclaration en C Il peut prendre deux états VRAI ou FAUX .

Les principaux types de variables :

2.2.2 Expression

Définition: En combinant des noms de variables, des opérateurs, des parenthèses et des appels de fonctions on obtient des expressions. Une règle simple 'a retenir :
En C++, on appelle expression tout ce qui a une valeur.

Operateurs arithmétiques:

+ : addition,
- : soustraction,
* : multiplication,
/ : division. Attention : entre deux entiers, donne le quotient entier,
Exemples :
19.0 / 5.0 vaut 3.8,

Type	Taille (En bits)	Signé	Non Signé (Unsigned)
Caractère → `Char`	8	-128 à +127	0 à +255
Entier → `Int`	16	-32768 à +32767	0 à +65535
Entier Court → `Short`	16	-32768 à +32767	0 à +65535
Entier long → `Long`	32	-2147483648 à +2147483647	0 à +4294967295
Réel → `Float`	32	+/- 3,4*10^{-38} à +/- 3,4*10^{+38}	Aucune Signification
Réel double précision → `Double`	64	+/- 1,7*10^{-308} à +/-1,7*10^{+308}	Aucune Signification

FIGURE 2.7

19 / 5 vaut 3,
19

Opérateurs incrémentation ou décrémentation:
Les affectations les plus fréquentes sont du type:
I = I + 1 et I = I − 1 .
En C, nous disposons de deux opérateurs inhabituels pour ces affectations :
Exemple opérateur avant: Exemple opérateur (++) ou (–) après:

Operateurs d'affectation:
= (égal) Forme générale de l'expression d'affectation : $\langle variable\rangle = \langle expression\rangle$.

Fonctionnement :

l'$\langle expression\rangle estd'abordévaluée; cettevaleurdonnelavaleurdel'expressiond'affectation.$

effet de bord : la $\langle variable\rangle$ *reçoit ensuite cette valeur.*
$Formegénérale : \langle variable\rangle\langle opérateur\rangle = \langle expression\rangle$

I++ ou ++I	pour l'incrémentation	(augmentation d'une unité)
I-- ou --I	pour la décrémentation	(diminution d'une unité)

FIGURE 2.8

L'expressionestéquivalente‘a :
$\langle variable\langle = \langle variable\rangle\langle opérateur\rangle\langle expression\rangle$

Par exemple, l'expression i += 3 équivaut ‘a i = i + 3.

2.2.3 Instructions

Une instruction est la définition d'une action , qui lorsqu'elle exécutée , modifie l'état d'un programme . On peut identifier comme instructions de base en algorithmique :

Instruction-expression
Forme : $\langle expression\rangle$;

```
#include <stdio.h>
/* Déclaration de variable a et b */
int a,b;

void main()
{
    /* Initialisation de a et b */
    a=2;
    b=3;
    /* Affichage de a avec incrémentation avant l'utilisation */
    printf("a = %d\n",++a);
    /* Affichage de b avec décrémentation avant l'utilisation */
    printf("b = %d",--b);
}
```

Figure 2.9

Cette instruction n'est utile que si l'$\langle expression \rangle auneffetdebor$.
$Exemples$: $Instruction - bloc$:
$Forme$: $\langle déclarationsetinstructions \rangle$

2.2.3.1 Traitements conditionnels

Elles permettent en fonction d'une condition, de choisir de faire une instruction ou un bloc d'instructions plutôt qu'un autre.
La structure **SI ... ALORS ...**
C'est la structure de base.
Exemple: La structure **SI ... ALORS ... SINON**.
Si la condition est vraie alors elle exécute l'instruction ou le bloc d'instructions qui suit le if (si),
par contre si la condition est fausse alors elle exécute l'instruction ou le bloc d'instructions qui suit else (sinon). **Exemple:** La structure de choix:
Elle permet en fonction de différentes valeurs d'une variable de faire plusieurs actions,

```
#include <stdio.h>
/* Déclaration de variables a et b */
int a,b;

void main()
{
    /* Initialisation de a et b */
    a=2;
    b=3;
    /* Affichage de a avec incrémentation après l'utilisation */
    printf("a = %d\n",a++);
    /* Affichage de b avec décrémentation après l'utilisation */
    printf("b = %d\n",b--);

    /* Affichage de a */
    printf("a = %d\n",a);
    /* Affichage de b */
    printf("b = %d",b);
}
```

FIGURE 2.10

si aucune valeur n'est trouvée alors ce sont les instructions qui suivent default qui sont exécutées.

exosup.com

Exemple :

2.2.3.2 Boucles

definition: Une structure itérative ou boucle permet d'exécuter plusieurs fois de suite une même séquence d'instructions. Cet ensemble d'instructions s'appelle le corps de la boucle. Chaque exécution du corps d'une boucle s'appelle une itération.
Il existe trois types de boucle :

While
do ... while
for

La structure **tant que , while :**
Dans cette structure la condition est testée au début.

exosup.com page facebook

Opérateur	équivalent	Notation
+=	x = x + y	x += y
−=	x = x − y	x −= y
*=	x = x * y	x *= y
/=	x = x / y	x /= y
%=	x = x % y	x %= y
=	x = x y	x = y
<<=	x = x << y	x <<= y
&=	x = x & y	x &= y
|=	x = x | y	x |= y
^=	x = x ^ y	x ^= y

Exemple : La structure **FAIRE ... TANT QUE , do ... while**
Dans cette structure la condition est testée à la fin.
Exemple: La structure **POUR ... FAIRE ... JUSQU'A ... for :**

Dans cette structure la condition est testée au début. Elle est très intéressante car elle est composée de trois parties:
L'instruction ou plusieurs instructions d'initialisation qui sont exécutées une seule fois au début de la structure.
L'instruction ou le bloc d'instructions du corps de la structure qui est exécuté à chaque itération.
L'instruction ou plusieurs instructions qui sont exécutées à la fin de chaque itération.

Exemple : beginfigure[!h]

```
Syntaxe:
if (condition) instruction;

ou

if (condition)
{
      instruction1;
      ...
      instructionn;
}
```

Figure 2.12

```
#include <stdio.h>
int i;
void main()
{
        for(i=0;i<10;i++)

                printf("i=%d\n",i);
}
```

```
#include <stdio.h>

int a,b;

void main()
{
    /* Saisie de a  et de b */
    printf("Donnez les valeurs de a et de b ");
    scanf("%d %d",&a,&b);

    /* Structure SI ALORS */
    if (a>b)
    {
            printf("a=%d est supérieur à b=%d\n",a,b);
            printf("\n");
    }
}
```

```
Syntaxe:
if (condition) instruction1; else instruction2;

ou

if (condition)
{
      instruction1;
      ...
}
else
{
      instruction2;
      ...
}
```

Figure 2.13

```
#include <stdio.h>

int a,b;

void main()
{
    /* Saisie de a  et de b */
    printf("Donnez les valeurs de a et de b ");
    scanf("%d %d",&a,&b);

    /* Structure SI ALORS SINON */
    if (a>b)
    {
            printf("a=%d est supérieur à b=%d",a,b);
            printf("\n");
    }
    else
    {
            printf("a=%d est inférieur ou égal à b=%d",a,b);
            printf("\n");
    }
}
```

exosup.com

Figure 2.14

exosup.com page facebook

```
Syntaxe:
switch (identificateur)
{
case valeur1 :
        instruction_1 ;         ou        {instructions_1...} ;
     break;
case valeuurm :
        instruction 2 ;         ou        {instructions 2...} ;
     break;
case valeurk:
        instruction_j ;         ou        {instructions_j...} ;
     break;
default :
        instruction i ;         ou        {instructions i...} ;
     break;
}
```

Figure 2.15

```c
#include <stdio.h>

char choix;

void main()
{
    /* Affichage du menu */
    printf("\n\n\n\n\n");
    printf("\t\t\t\t MENU\n");
    printf("\t a --> ACTION 1\n");
    printf("\t b --> ACTION 2\n");
    printf("\t c --> ACTION 3\n");
    printf("\t d --> ACTION 4\n");
    printf("\n\n\t\t tapez sur une touche en minuscule  ");

    /* saisie de la touche */
    choix=getchar();

    /* Structure de choix switch*/
    switch(choix)
    {
            case 'a':{
                        printf("Exécution de l'ACTION1");
                        printf("\n");
                     } break;
            case 'b':{
                        printf("Exécution de l'ACTION2");
                        printf("\n");
                     } break;
            case 'c':{
                        printf("Exécution de l'ACTION3");
                        printf("\n");
                     } break;
            case 'd':{
                        printf("Exécution de l'ACTION4");
                        printf("\n");
                     } break;
            default :{
                        printf("Mauvaise touche, pas d'ACTION");
                        printf("\n");
                     }
    }
}
```

Figure 2.16

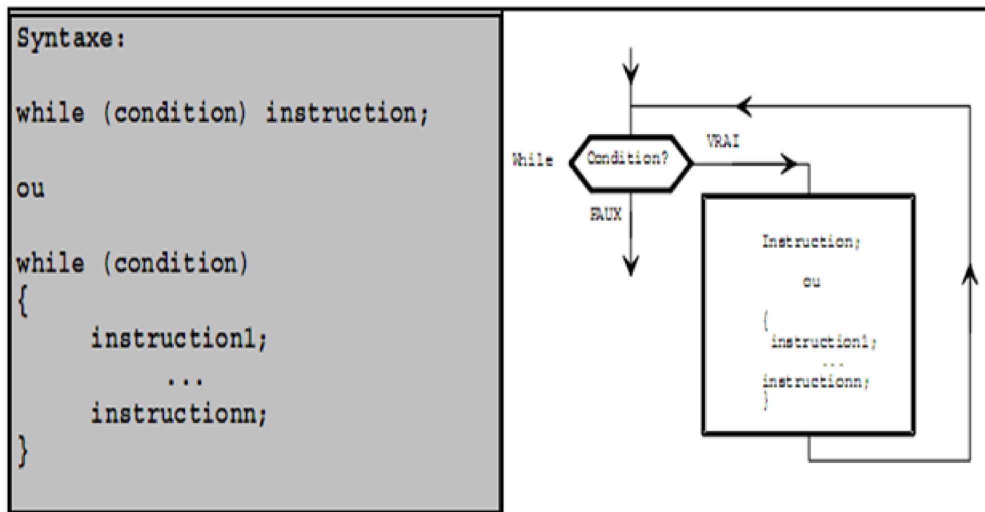

Syntaxe:
while (condition) instruction;
ou
while (condition)
{
instruction1;
...
instructionn;
}
While
Condition?
VRAI
FAUX
Instruction;
ou
{
instruction1;
...
instructionn;
}

Figure 2.17

```
#include<stdio.h>
int i ;
void main()
{
/* boucle while<tant que faire> */
i = 0 ;
while ( i<10 )
        {
        printf ("i = %d\n ",i) ;
        i++ ;
        }
}
```

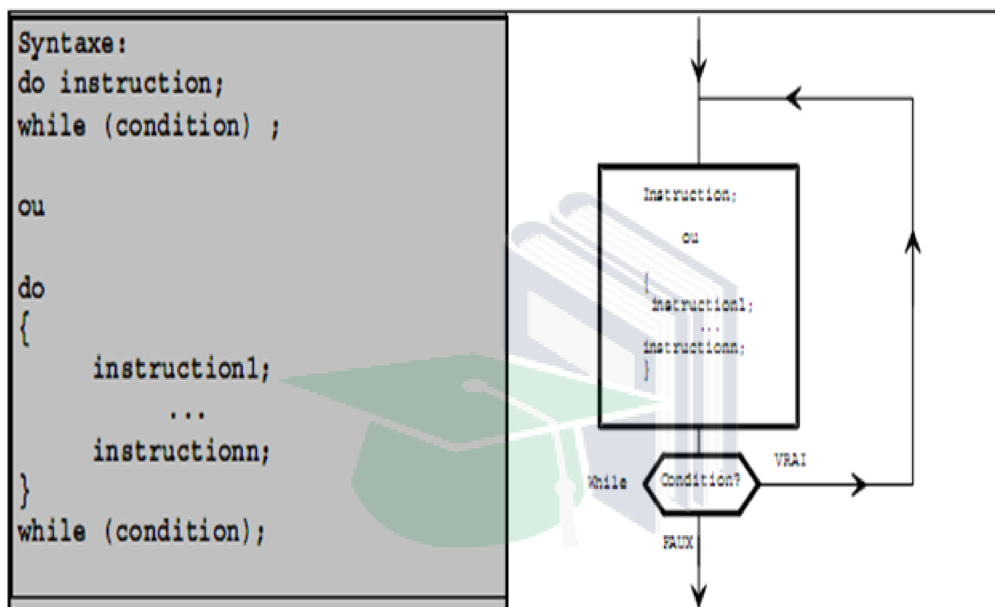


Figure 2.18

exosup.com page facebook

```
#include <stdio.h>
int i;
void main()
{
i=0;
/* Boucle while <faire tant que> */
do
      {
      printf("i= %d\n",i);
      i++;
      }
while(i<10);
}
```

Chapitre 3

Applications aux instruments d'optiques

3.1 Structure d'un oeil normal

Un oeil normal est constitué de: la chambre antérieure remplie d'humeur aqueuse, limitée d'un côté par la cornée, et de l'autre par le cristallin. une lentille convexe complexe: le cristallin, qui n'est pas un milieu homogène mais est constitué d'une succession de "couches" comme indiqué sur la figure. Le muscle ciliaire, qui fait partie du corps ciliaire, sert à déformer le cristallin, changeant ainsi sa distance focale. Le cristallin est diaphragmé par la pupille, d'ouverture variable. une cavité qui occupe la plus grande partie du volume de l'oeil et est remplie par l'humeur vitrée La rétine sur laquelle se projette l'image fournie par le cristallin Un oeil normal a une profondeur d'environ 24 mm. Deux éléments de ce système méritent quelques précisions :L'oeil a une distance focale variable qui permet de former sur la rétine (dont la distance au cristallin est constante) l'image d'objets situés à des distances variables. Entre quelles valeurs extrêmes cette distance focale varie-t-elle ? L'oeil a une distance focale maximale lorsque les muscles ciliaires sont au repos: on dit que l'oeil n'accommode pas. Dans ces conditions, on peut voir des objets situés à une distance maximale appelée "punctum remotum", qui est infinie pour un oeil normal. Lorsque la distance focale de l'oeil est minimale, il peut voir nettement des objets situés à une distance minimale appelée "punctum proximum", qui est d'environ 25 cm pour un oeil normal moyen (en fait, ce "punctum proximum" varie beaucoup avec l'âge : plus faible chez les enfants, il s'accroît régulièrement au fil des ans).Remarque :nous parlons de la "distance focale de l'oeil" plutôt que de celle du cristallin seulement: en effet, la distance focale du système optique constitué par l'oeil dépend non seulement de la

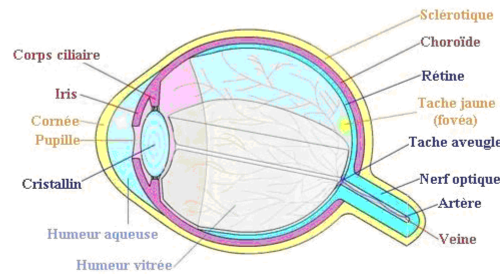


Figure 3.1

forme du cristallin, mais aussi de celle de la cornée: la lumière traverse une surface de séparation entre l'air et l'humeur aqueuse : la cornée, avant d'arriver sur le cristallin. L'effet de lentille est donc obtenu grâce au passage de nombreux dioptres (et non de deux seulement comme dans le cas d'une lentille ordinaire): essentiellement, il s'agit des deux dioptres traversés au passage de la cornée, de l'interface humeur aqueuse/cristallin, et de l'interface cristallin/humeur vitrée, sans oublier que le cristallin lui-même n'est pas homogène, et que sa traversée correspond donc à la traversée de plusieurs dioptres successifs.Comment l'image formée sur la rétine est-elle "lue" par le cerveau ? La rétine est très différente des écrans diffusants utilisés dans les manipulation d'optique. La rétine ne rediffuse pas l'image vers un observateur, mais la transforme en signal transmis au cerveau via le nerf optique grâce à un ensemble de capteurs, les cônes et les bâtonnets, qui ne sont pas répartis de façon uniforme sur toute la rétine. La plus grande densité de capteurs correspond à une petite zone de la rétine: la fovéa, sur laquelle se concentrent l'essentiel des cônes, qui permettent la vision des couleurs (pour plus de précisions sur la vision des couleurs voir la fiche sur "La trivariance visuelle chez l'homme" sur le site CultureSciences-Physique). En outre, sur la fovéa, une cellule réceptrice correspond à une fibre nerveuse: d'où une grande résolution dans cette zone de la rétine. Lorsqu'on s'éloigne de la fovéa, la vision est plus floue. Une même fibre nerveuse est reliée à un grand nombre de bâtonnets.Cette disposition (grand nombre de cellules réceptrices par fibre nerveuse) est avantageuse de nuit, puisqu'elle permet une grande sensibilité de la rétine à des intensités lumineuses réduites: les intensités lumineuses reçues par chacun des bâtonnets reliés à une même fibre nerveuse s'additionnent.

Les éléments et propriétés qui permettent la vision peuvent ainsi se décomposer en quatre points principaux :

• un système optique, succession de dioptres, plus complexe dans sa structure qu'une lentille ordinaire. Pour comprendre le principe du fonctionnement de l'oeil, ce système

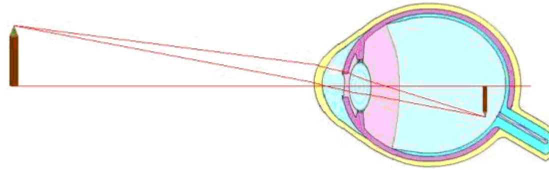


FIGURE 3.2: Un oeil myope forme l'image d'un objet très éloigné avant la rétine

est souvent modélisé par une simple lentille
• ce système est adaptable. Lorsque un objet se déplace, son image par une lentille ordinaire se déplace et il faut donc déplacer l'écran diffusant pour récupérer cette image. Dans le cas de l'oeil, l'"écran" ne bouge pas mais c'est la "lentille" qui s'adapte. De plus, l'oeil possède un diaphragme: la pupille, qui s'adapte automatiquement à l'éclairement.
• un système de capteurs non uniformément répartis sur la rétine, et de natures différentes
• qui est en fait un prolongement du cerveau qui va récupérer l'information visuelle via le nerf optique puis l'interpréter

3.2 Les défauts du système optique

Un certain nombre de défauts de la vision (myopie, hypermétropie, et astigmatisme) sont des défauts du système optique de l'oeil: soit ce dernier n'est pas assez convergent, soit il est trop convergent, soit il a une forme irrégulière, soit encore le système optique cristallin + rétine est normal, mais c'est l'oeil qui est de dimension inhabituelle.

3.2.1 A- L'oeil myope et l'oeil hypermétrope

L'oeil myope peut être un oeil dont la "lentille" est trop convergente (distance focale au repos trop courte), ce qui fait que l'image d'un objet à l'infini se forme avant la rétine. Alternativement, l'oeil myope peut être un oeil trop grand (distance cristallin-rétine trop importante) avec une "lentille" normale, ce qui donne le même résultat.
Un oeil myope ne peut ainsi pas voir nettement un objet situé à l'infini: son punctum remotum est inférieur à l'infini. Par contre, si ses capacités d'accommodation sont

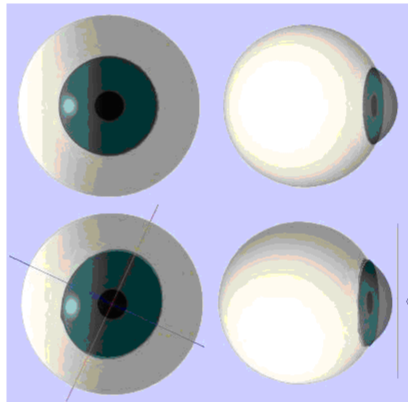


FIGURE 3.3: Un oeil hypermétrope, s'il n'accommode pas, forme l'image d'un objet très éloigné après la rétine

normales, il peut voir des objets placés très près: son punctum proximum est plus faible que celui d'un oeil normal. L'oeil hypermétrope peut être un oeil dont la "lentille" est trop peu puissante (distance focale au repos trop grande), ce qui fait que l'image d'un objet à l'infini, lorsque l'oeil n'accommode pas, se forme après la rétine. Alternativement, l'oeil hypermétrope peut être un oeil trop petit (distance cristallin-rétine trop faible) avec une "lentille normale", ce qui donne le même résultat. Un oeil hypermétrope doit ainsi accommoder pour voir nettement un objet situé à l'infini. Son punctum remotum reste infini, mais s'il n'accommode pas, il voit flou. S'il possède des capacités d'accommodation moyennes, la distance focale minimale (à accommodation maximale) de l'oeil hypermétrope, c'est-à-dire la valeur de son punctum proximum, est plus grande que celle d'un oeil normal: il voit flou des objets proches qu'un individu normal ou myope voit nettement. Degré de myopie ou d'hypermétropie de l'oeil, correction par des lunettes ou des lentilles: Le degré de myopie ou d'hypermétropie d'un oeil est donné par la valeur de la correction qu'il faut apporter à cet oeil pour qu'il voie normalement. Ainsi, un oeil myope avec un punctum remotum à p mètres, doit être corrigé par une lentille divergente qui fera d'un objet à l'infini une image à p mètres: la vergence correspondante est C = 1/f' = 1/—p—. Un oeil qui ne voit net que jusqu'à 40 cm = 0,4 m doit être corrigé par des verres de - 1/0,4 = - 2,5 dioptries. Un oeil hypermétrope qui ne voit net qu'au-delà d'une distance de p' cm au lieu des 25 cm correspondant au punctum proximum d'un oeil normal doit être corrigé par une lentille convergente qui fait d'un objet situé à 0,25 m une image située à p' m: la relation de conjugaison donne: 1/-p' - 1/-0,25 = C soit C = 4 - 1/p'. Un oeil hypermétrope qui ne voit net qu'au-delà de 40 cm correspondra donc à une correction de 4 - 1/0,4 = 1,5 dioptries. Remarque : Un oeil hypermétrope, si l'hypermétropie est suffisamment légère, peut compenser son défaut par un surcroît d'accommodation : il peut voir net à l'infini, il lui suffit d'accommoder là où un oeil normal n'a pas besoin

exosup.com page facebook

de le faire. Un oeil myope ne peut en aucun cas voir net de loin : l'accommodation, qui diminue la distance focale du système optique de l'oeil, rapproche en effet l'image du cristallin. Une conséquence de cette particularité est que l'hypermétropie non corrigée peut entraîner des migraines, ce qui n'est pas le cas de la myopie. Une hypermétropie légère peut en outre passer inaperçue jusqu'à ce que la presbytie (voir paragraphe 2.1.) vienne diminuer la puissance d'accommodation de l'oeil.

3.2.1.1 B- L'oeil astigmate

Une lentille "idéale" est composée de dioptres parfaitement sphériques. Si ce n'est pas le cas, on observe des erreurs d'astigmatisme. Un oeil astigmate possède généralement une cornée en forme de "ballon de rugby" plutôt qu'en forme de "ballon de football". Cette déformation est caractérisée par deux axes, généralement perpendiculaires, caractéristiques de la déformation.

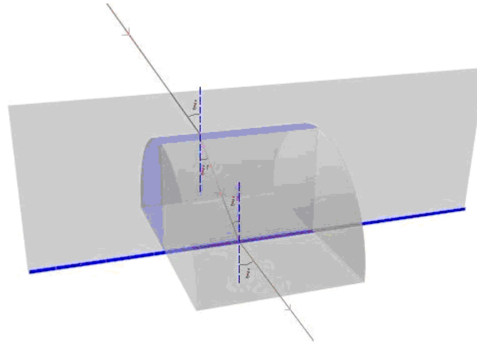


FIGURE 3.4

Schéma en haut à gauche : oeil normal vu de face, en haut à droite : oeil normal de profil. Schéma en bas à gauche : oeil astigmate vu de face, l'axe rouge correspond à l'axe suivant lequel la courbure de la cornée est modifiée par rapport à celle d'un oeil normal, l'axe bleu correspond à l'axe suivant lequel la courbure de la cornée est normale. En bas à droite : l'oeil astigmate précédent vu suivant l'axe bleu (on voit donc la cornée dans le plan dans lequel elle est la plus déformée) Dans le cas de l'astigmatisme simple (comme ci-dessus), la courbure de la cornée n'est déformée par rapport à la normale que suivant un axe (l'axe rouge sur le dessin) et il existe un axe (axe bleu) suivant lequel elle est normale. Dans ce cas, l'image d'un point à l'infini n'est plus ponctuelle : une de ses extrémités est située, sans accommodation, sur la rétine, tandis que l'autre est située en-dehors de la rétine. Par exemple, dans le cas représenté sur la figure ci-dessus, la cornée est normale suivant l'axe bleu, et trop peu convergente (courbure plus faible) suivant l'axe rouge : l'"image" (non-ponctuelle) par cet oeil d'un

point situé à l'infini aura alors, en l'absence d'accommodation, une extrémité sur la rétine et une autre derrière la rétine. On parle d'astigmatisme hypermétropique simple. Si la cornée est au contraire trop convergente suivant l'axe rouge, et normale suivant l'axe bleu, on parle d'astigmatisme myopique simple. Astigmatisme composé : des situations plus complexes peuvent se produire : tout d'abord, on peut avoir une cornée trop convergente suivant les deux axes, mais de courbure différente selon l'axe (N.B.: dans le cas de la myopie simple, la cornée est trop convergente mais elle n'est pas asymétrique), on parle d'astigmatisme myopique composé. De façon similaire, la cornée peut être trop peu convergente suivant les deux axes, mais de courbure là encore différente selon l'axe, on parle alors d'astigmatisme hypermétropique composé. Enfin, la situation la plus complexe est quand la cornée est trop convergente suivant un axe, trop peu suivant l'autre : dans ce cas, on a un astigmatisme mixte. Correction par des lunettes ou des lentilles : Les lentilles permettant de corriger l'astigmatisme sont elles-mêmes astigmates: leur non-sphéricité est calculée pour compenser celle de l'oeil. Dans le cas d'un astigmatisme simple, on doit utiliser une lentille qui ne dévie pas les rayons lumineux qui passeront suivant l'axe bleu (axe suivant lequel la cornée a une courbure normale) mais déviera ceux qui passent suivant l'axe rouge de courbure anormale de la cornée. La solution à ce problème consiste à choisir une lentille dite "cylindrique" en forme de cylindre tronqué :

Un rayon lumineux passant par le plan perpendiculaire au plan de coupe du cylindre et passant par l'axe de la lentille (plan bleuté) n'est pas dévié après passage par la lentille (la normale au dioptre d'entrée du rayon lumineux est parallèle à la normale au dioptre de sortie, comme dans le cas d'une lame à faces parallèles). Un rayon lumineux passant hors de ce plan est dévié. On placera donc l'axe (bleu) de la lentille hémi-cylindrique parallèlement à l'axe suivant lequel la cornée a une courbure normale (axe bleu sur le dessin). Cet axe est l'axe que l'on appelle traditionnellement "l'axe d'astigmatisme". On donnera donc pour un oeil affecté d'un astigmatisme simple la direction de l'axe d'astigmatisme de l'oeil (donnée par rapport à l'horizontale, -25 environ dans le cas de notre dessin) ainsi que le nombre de dioptries correspondant à la correction à apporter suivant l'axe rouge (axe de plus forte déformation). Dans le cas d'un astigmatisme hypermétropique simple, la correction sera un nombre positif de dioptries (par exemple, on aura (-25, +2 dioptries), ou, écrit plus simplement, (-25, +2)) dans le cas d'un astigmatisme myopique simple, la correction sera un nombre négatif (par exemple, (150, -1,5 dioptries) ou, écrit plus simplement (150, -1,5)). L'astigmatisme composé, comme il correspond à deux courbures trop grandes ou trop petites, mais différentes, sera associé à deux nombres de dioptries, par exemple (90, +2) +1,5 dans le cas d'un astigmatisme hypermétropique composé, ou (120, -3) -4 dans le cas d'un astigmatisme myopique composé.

Un astigmatisme mixte, quant à lui, combinera un nombre positif et un nombre négatif de dioptries : par exemple (-30, -0,5) +1.* *Remarque : lors d'une consultation chez un ophtalmologiste, on aura une paire de données de ce genre, une pour l'oeil gauche, et une pour l'oeil droit, l'ophtalmologiste écrira donc quelque chose qui ressemblera à : OD = (50, +2) +1,5 , OG = (35, +3) +2,5

3.2.1.2 C- Problèmes de transparence

Le système optique de l'oeil, pour bien fonctionner, doit avoir une forme et une puissance adéquate, mais il doit aussi évidemment laisser passer la lumière... La cornée, l'humeur aqueuse, le cristallin et le vitré sont donc tous, dans un oeil normal, parfaitement transparents. Une opacification d'une de ces parties (taies cornéennes, opacification du cristallin, "flotteurs" opaques dans le vitré) peut cependant toujours se produire et se traduire par des troubles plus ou moins importants de la vision. La cataracte est l'opacification du cristallin. Elle survient le plus souvent chez les personnes âgées. Peu importante et survenant à un âge avancé, elle est parfois simplement ignorée car elle n'entraine pas de problème important. Plus importante, on traite la partie opacifiée du cristallin. Dans le pire des cas, une cataracte congénitale ou très avancée peut se traiter en enlevant carrément le cristallin opacifié, qui peut être remplacé par un implant. En l'absence de cristallin, l'oeil a deux défauts majeurs: très forte hypermétropie (du fait de l'absence de la lentille que constitue le cristallin), et absence d'accommodation. Un implant constitué par une "simple" lentille de même vergence que le cristallin peut corriger l'hypermétropie, mais ne peut pas résoudre totalement le deuxième problème: le cristallin artificiel inséré n'a pas les propriétés d'adaptabilité du cristallin de départ: l'oeil a donc une capacité d'accommodation nulle ou très faible (si le cristallin artificiel a une certaine souplesse permettant un minimum d'accommodation); il est semblable en cela à un oeil atteint d'une presbytie extrême.

3.3 Les défauts d'adaptation

3.3.1 A- L'oeil presbyte

Lorsque l'oeil vieillit, il a tendance à devenir presbyte.
La presbytie se caractérise par la perte progressive du pouvoir d'accommodation de l'oeil. Par conséquent, le punctum proximum d'un presbyte se rapproche de plus en plus de son punctum remotum (qui correspond à la distance de vision nette pour un oeil qui n'accommode pas): dans la pratique, la personne doit de plus en plus éloigner

les textes qu'elle lit. La presbytie se corrige pour la vision de près par le port de verres convergents, qui ne sont pas nécessaires par contre pour la vision de loin.
Un myope n'aura pas besoin, pendant un certain temps, de corriger sa vision de près qui était meilleure que la moyenne (mais sa vision de loin ne s'améliorera par contre pas). Dans certains cas, la presbytie peut révéler une hypermétropie passée inaperçue: une hypermétropie légère peut en effet être compensée par un effort d'accommodation supplémentaire, mais avec l'âge, cette capacité d'accommodation diminue et l'hypermétropie se révèle. La presbytie semble alors plus précoce que la moyenne.

3.3.2 B- Lorsque vos "lunettes de Soleil" n'en sont pas

La pupille permet d'adapter à la luminosité extérieure la quantité de lumière que votre oeil laisse passer. En cas de grande luminosité, la pupille diminue de taille, évitant l'éblouissement et améliorant la netteté de la vision. Lorsque l'éblouissement est trop conséquent, la pupille ne suffit plus à protéger les yeux, c'est pour cela que l'on porte des lunettes de soleil. Cependant des lunettes de soleil de mauvaises qualités peuvent avoir un effet dramatique: si vos lunettes de Soleil ne sont pas de véritables lunettes prévues pour protéger vos yeux des UV solaires, mais seulement des verres teintés, porter ces lunettes peut être encore plus dangereux que de ne rien porter. En effet, des verres teintés qui diminuent la luminosité dans le visible amènent la pupille à se dilater, laissant ainsi passer plus de lumière et en particulier plus de rayonnement ultraviolet dangereux pour la rétine... par conséquent si vos verres teintés diminuent la luminosité dans le visible mais ne sont pas conçus pour protéger efficacement des UV, votre rétine reçoit encore plus d'ultraviolets lorsque vous les portez que si vous ne les mettez pas!!...

3.4 Les problèmes de la rétine

3.4.1 A- Le daltonisme

La présence de trois types de cônes différents de maximums de sensibilité respectivement dans le bleu-violet, le vert, et le jaune-vert, permet une vision des couleurs normale chez l'homme (cf la fiche sur "La trivariance visuelle chez l'homme" sur le site CultureSciences-Physique). Si l'une des espèces de cônes est absente, la vision des couleurs est altérée, c'est le daltonisme. Il se peut aussi qu'une des espèces de cônes ne soit pas absente mais déficiente. Cette anomalie, moins grave, altère aussi la vision des couleurs. Le daltonisme, et les défauts de vision des couleurs analogues, sont des affections héréditaires qui touchent quelques 8Cette différence de pourcentage a pour origine la transmission du daltonisme par le chromosome X.

B- La dégénérescence maculaire et la rétinite

La dégénérescence maculaire et la rétinite sont toutes deux des affections de la rétine, correspondant à la destruction progressive des capteurs dans certaines zones de la rétine. Dans le cas de la dégénérescence maculaire, la macula, c'est-à-dire la zone où se trouve la fovéa, est touchée en premier, entraînant une perte de la vision centrale. Dans d'autres cas, c'est la vision périphérique qui est touchée et amène une vision "tunnel". Ces affections peuvent être liées à l'âge (dégénérescence maculaire liée à l'âge) mais aussi au diabète ou à des agressions par le soleil, etc.

3.4.2 C- Les limites inhérentes à la nature de la rétine

La rétine ne peut pas résoudre des intensités lumineuses plus faibles que l'intensité minimale nécessaire pour exciter les bâtonnets et ne peut pas percevoir non plus des détails plus petits que la distance entre deux capteurs. Ces deux limitations ont des impacts différents sur la qualité de notre vision. En effet, la première "limitation" n'en est quasiment pas une, car, dans la quasi-obscurité, l'oeil peut percevoir une lumière correspondant à quelques photons seulement, sa sensibilité à la lumière est donc très importante. La deuxième limitation, par contre, peut être mise en évidence par une expérience simple (site CultureSciences-Physique) grâce à laquelle vous pouvez mettre en évidence la distance moyenne entre deux capteurs dans un oeil normal moyen : on trace deux lignes parallèles sur une feuille, et on se place à des distances variables de la feuille. Au-delà d'une certaine distance, on ne peut plus distinguer les deux lignes. La distance ainsi obtenue permet, pour un oeil normal ou bien corrigé par une paire de lunettes, de trouver l'ordre de grandeur de la distance entre deux capteurs de l'oeil.

- Conclusion

Les problèmes de la vision sont multiples, et nous n'avons fait qu'effleurer le sujet. Myopie, hypermétropie, astigmatisme, presbytie, sont des sujets sur lesquels nous

exosup.com page facebook

Annexe A

Manuel de Latex

A.1 Introduction

Ce petit manuel vous permet de prendre en main le logiciel LaTeX. Selon le dicton "un exemple vaut mieux qu'un long discours", on propose de donner quelques exemples d'insértion des tableaux, figures, equations, etc. Le code LaTeX de chaque exemple est donné dans le fichier AnnexeA.tex.

NB/ Les fichiers Latex.pdf, symbols.pdf et ltxdoc.pdf donnent des exemples concrets sur les points saillants mentionnés précédemment (les Equations, insertion des Figures, Tableaux, les Listes, etc.).

L'ajout des références, se fait au niveau du fichier Bibliography.bib. Pour gérer/ajouter des références, il faut installer l'appliaction Jabref qui est un logiciel de gestion bibliographique libre.

Pour citer une référence, utiliser la commande `\cite{Reference2}` : [**?**]

Utiliser l'instruction `\textbf{texte}` pour mettre **le texte en gras**

Se référencer à la page web www.commentcamarche.net/contents/625-latex-mise-en forme pour de plus amples informations sur la Mise en forme en Latex,

A.2 Insértion des caractères spéciaux

Pour insérer un symbole dans le texte, il faut mettre son nom entre deux symboles `$`.

Exemples :

- `$\Delta$` donne Δ.
- `$\overline{abc}$` donne $\overline{abc}$.
- $f(z) = \begin{cases} \overline{\overline{z^2} + \cos z} & \text{for} \quad |z| < 3 \\ 0 & \text{for} \quad 3 \leq |z| \leq 5 \\ \sin \overline{z} & \text{for} \quad |z| > 5 \end{cases}$

A.3 Les Listes

A.3.1 Listes simples

Pour insérer une liste simple, il faut utiliser l'instruction `\begin{itemize} ... \end{itemize}`. chaque élément d'une liste devant commencer par `\item`.

Exemple :

- texte1
- texte2
- texte3

A.3.2 Listes numérotées

Pour insérer une liste simple, il faut utiliser l'instruction `\begin{enumerate} ... \end{enumerate}`. chaque élément d'une liste devant commencer par `\item`.

Exemple :

1. texte1
2. texte2
3. texte3

A.4 Les Tableaux

Pour insérer un tableau, il faut utiliser l'environnement tabular qui s'utilise de la manière suivante :

```
\begin{tabular}{pos}
champ1 & champ2... \\ % ligne 1
champ1 & champ2... \\ % ligne 2
:
\end{tabular}
```

pos	définit la position du texte dans chaque colonne, l pour left, c pour center, r pour right;
`&`	définit le passage à la colonne suivante ;
`\\`	définit un passage à la ligne suivante.

Quelques exemples :

A.4.1 Exemple simple d'un tableau avec colorisation des cellules:

la commande `\rowcolor{color}` permet de coloriser les cellules. color prend les valeurs, black, white, red, green, blue, cyan, magenta, yellow. A déplacer dans la cellule de votre choix.

Champ 1	Champ2
Champ3	Champ4
Champ5	Champ5.

TABLE A.1: Ecrire Votre légende ici.

A.4.2 Exemple d'un tableau avec fusion de cellules

L'instruction `\multicolumn` permet de fusionner les colonnes alors que l'instruction `\multirow` fusionne les lignes.

ligne1 champ1	champ2	champ3	1.23
ligne2 champ1	champ2 + champ3		12.3
ligne3 champ1	champ2	champ3	12.34

TABLE A.2: Ecrire Votre légende ici.

Methods	Parameters	Mean value of VINET
SRM	Q= 600	85%
	Q= 800	85,6%
	Q= 1500	87,5%
	Q= 3000	86%
CSC	t=5	67,5%
	t=8	82%
	t=12	81,5%
	t=15	78,5%

TABLE A.3: Mean value of VINET criterion.

A.4.3 Exemple d'un tableau avec changement de la taille des cellules

Par exemple, l'insertion M3cm donne une taille de 3cm à la cellule où elle est placée . Enlever la barre verticale dans un emplacement donné |M3cm, implique la suppression de la ligne correspondante dans le tableau. Ainsi, le tableau A.4 devient :

Gradient / Invariant	Carron	Dizenzo	MGradient	GradientC
RGB	71.3%	**83.5%**	61.6%	**83.8%**
Greyworld	75.2%	**85%**	63%	**84.7%**
MaxRGB	75.5%	**84.5%**	63.3%	**84.6%**
M-intensity	76%	**84%**	63.3%	**84.5%**
RGB-rang	61%	77%	59.5%	80.6%
N-affine	75%	**84.5** %	63.8%	**84.1** %
$c_1c_2c_3$	55%	60%	51.6%	52.4%
$m_1m_2m_3$	46%	54%	49%	52%
$l_1l_2l_3$	52%	47%	44.7%	25.5 %

TABLE A.4: Votre légende ici.

Gradient / Invariant	Carron	Dizenzo	MGradient	GradientC
RGB	71.3%	**83.5%**	61.6%	**83.8%**
Greyworld	75.2%	**85%**	63%	**84.7%**
MaxRGB	75.5%	**84.5%**	63.3%	**84.6%**
M-intensity	76%	**84%**	63.3%	**84.5%**
RGB-rang	61%	77%	59.5%	80.6%
N-affine	75%	**84.5** %	63.8%	**84.1** %
$c_1c_2c_3$	55%	60%	51.6%	52.4%
$m_1m_2m_3$	46%	54%	49%	52%
$l_1l_2l_3$	52%	47%	44.7%	25.5 %

TABLE A.5: Votre légende ici.

A.5 Les Figures

Les figures et les images doivent être stockées dans le réprtoire Img.

Etant donné que l'on ne dispose que de deux images (IT.png et Lastid.png), il est donc naturel qu'elles soient rencontrées dans les exemples donnés ci-dessous.

A.5.1 Insértion d'une simple figure

La figure A.1 illustre le logo de l'Université Ibn Tofail. Les instructions width et height permettent de modifier resepectivement la largeur et la hauteur de la figure.



FIGURE A.1: Logo de Ibn Tofail

A.5.2 Insértion des figures dans un tableau

Insertion des figures dans un tableau (cf. Figures A.2 et A.3).



FIGURE A.2: Insertion des figures dans un tableau

A.5.3 Placement des figures côte à côte

Placement des figures côte à côte avec l'instruction minipage (cf. Figures A.4)

A.6 Les Equations

Pour insérer une liste simple, il faut utiliser l'instruction `\begin{equation} ... \end{equation}`.

exosup.com page facebook



FIGURE A.3: Exemple 2 .



FIGURE A.4: Placement des figures côte à côte avec l'instruction minipage .

$$x = \frac{A}{B} \tag{A.1}$$

$$y = \sqrt{x^2} \tag{A.2}$$

$$\frac{1}{X_S} = \frac{4\pi}{\alpha^2} N \left[\frac{Z^{4/3}\ r_e^2}{A\beta^2} \right] \tag{A.3}$$

$$x = \begin{cases} y & \text{si } y > 0 \\ z + y & \text{sinon} \end{cases} \tag{A.4}$$

$$\lim_{t \to \infty} u(x,t) = \sqrt{\frac{2}{l}} \sum_{k=1}^{\infty} \left(\frac{l}{\pi k c} \right)^2 a_k \sin\left(\frac{\pi k}{l} x \right) \equiv v(x) \tag{A.5}$$

$$f(x) = \int \frac{\sin x}{x}\, \mathrm{d}x \tag{A.6}$$

$$f(x) = \prod_{i=1}^{n} \left(i - \frac{1}{2i} \right) \tag{A.7}$$

$$\left.\begin{array}{rrcrr} \text{a)} & y & = & c & (constant) \\ \text{b)} & y & = & cx + d & (linear) \\ \text{c)} & y & = & bx^2 + cx + d & (square) \\ \text{d)} & y & = & ax^3 + bx^2 + cx + d & (cubic) \end{array}\right\} \text{Polynomes} \tag{A.8}$$

L'environnement eqnarray permet d'aligner une formule sur 3 colonnes, à gauche. On passe d'une colonne à l'autre par `&` et d'une ligne à l'autre par `\\` . Chaque ligne est numérotée sauf si `\\` est précédé de la commande `\nonumber` (si l'on ne veut aucun n^o, on utilise eqnarray*).

$$u_t - c^2 u_{xx} = g(x,t), \tag{A.9}$$

$$u(x,0) = 0,$$

$$u_x(0,t) = u_x(l,t) = 0. \tag{A.10}$$

$$y = d$$

$$y = cx + d$$

$$y = bx^2 + cx + d$$

$$y = ax^3 + bx^2 + cx + d \tag{A.11}$$

left	middle	right
$\frac{1}{\sqrt{n}} =$	$\frac{\sqrt{n}}{n} =$	$\frac{n}{n\sqrt{n}}$

Encadrement [o]u

1ere ligne
2eme ligne
...

beginflushleft

A.6.1 fonction

Comme dans la plupart des langages, on peut en C découper un programme en plusieurs fonctions. Une seule de ces fonctions existe obligatoirement ; c'est la fonction principale appelée main. Cette fonction principale peut, éventuellement, appeler une ou plusieurs fonctions secondaires. De même, chaque fonction secondaire peut appeler d'autres fonctions secondaires ou s'appeler elle-même (dans ce dernier cas, on dit que la fonction est récursive).

Une fonction est un sous-programme qui reçoit éventuellement des arguments en entrée (ces arguments sont toujours des valeurs) et qui retourne éventuellement une valeur d'un certain type vers la fonction appelante.

Définition de fonction:
la syntaxe d'une fonction:

type-de-retour
nom-de-fonction(liste des arguments)

déclarations de variables locales
liste d'instructions
Les arguments d'une fonction sont simplement des variables locales qui sont initialisées par les valeurs obtenues lors de l'appel.
Les noms des arguments figurant dans l'en-tête de la fonction se nomment des ⋘ $argumentsmuets ⋙ ouencoreargumentsformelsouparamètresformelsLeurrôleestdepermettre, aus$
$Lesargumentsfournislorsdel'utilisation(l'appel)delafonctionsenommentdes ⋘ argumentseffectif$
$(ouencore ⋘ paramètreseffectifs ⋙).Lafonctionsetermineparl'instructionreturn:$

$return(expression); expressiondutypedelafonction$
$return; fonctionsanstype$

Exemple de fonction

```
int imax(int n, int m)
{
  int max;

  if (n>m)
          max = n;
  else
          max = m;

  printf(" le maximum de %d   et %d est  %d",m,n,max);
   return max;
}
```

Pour appeler une fonction, il suffit de mettre son nom suivi de la liste des valeurs à donner aux paramètres entre parenthèses. Un appel de fonction est considéré comme une expression du type du résultat de la fonction et la valeur de cette expression est le résultat de l'exécution de la fonction.
L'appel de fait par:
nom-fonction(liste des arguments);